

STATE SERVICES COMMISSION  
Te Komihana O Ngā Tari Kāwanatanga



---

# Guide to Legal Issues in Using Open Source Software

---

## Table of Contents

TABLE OF CONTENTS.....	2
PART A - INTRODUCTION .....	3
PART B - EXECUTIVE SUMMARY .....	3
PART C – BACKGROUND.....	5
KEY FEATURES OF OPEN SOURCE LICENCES .....	5
UNDERSTANDING THE INFECTIOUS EFFECTS OF OPEN SOURCE LICENCES .....	6
There are three broad levels of infection .....	7
Infectious licences can be quarantined by various integration techniques .....	9
LEGAL RISKS IN USING OPEN SOURCE SOFTWARE.....	10
There are a number of legal risks .....	11
Risks are affected by the software’s intended use .....	12
And the licence terms may never be enforced .....	12
PART D – RECOMMENDATIONS .....	14
USING STAND ALONE OPEN SOURCE APPLICATIONS.....	14
Only use specific open source licences.....	14
Obtain warranties/indemnities if appropriate and available .....	15
INTERNAL MODIFICATION AND INTEGRATION .....	16
Choose a distribution strategy.....	16
Manage the chosen licence to match the chosen strategy.....	17
Avoid conflicts of open source licences .....	19
THIRD PARTY SOFTWARE DEVELOPMENTS.....	19
Prohibit use of open source software without consent .....	19
If open source used and approved, include specific contractual provisions .....	20
SOFTWARE DISTRIBUTION .....	20
Confirm whether any open source licence applies.....	20
Meet the distribution requirements for all infected software .....	21
PART E - EXCEPTIONS .....	21
PART F – EXAMPLES .....	22
EXAMPLE ONE - STAND ALONE APPLICATION .....	22
EXAMPLE TWO - INTERNAL DEVELOPMENT .....	22
EXAMPLE THREE - EXTERNAL DEVELOPMENT .....	23

## PART A - INTRODUCTION

- 1 This guide was prepared by the State Services Commission (SSC) to assist New Zealand government agencies assess and mitigate the legal risks of using open source software.
- 2 Government agencies acquire open source software through a variety of channels, whether it be staff downloading open source code from the internet, or external developers providing software that includes open source components. While this use of open source software has many benefits, it brings with it a number of legal risks not posed by proprietary or commercial software. These include an increased risk of exposure to faults and intellectual property claims, and the risk of forced disclosure of confidential code.
- 3 There is no reason why agencies should not consider open source software on the same basis as commercial software. But agencies should base their decisions on the overall merits of the software concerned. This means weighing the unique legal risks of open source software together with the usual factors such as cost, functionality, interoperability and security.

## PART B - EXECUTIVE SUMMARY

- 4 Software is generally considered to be "open source" when it is provided in source code (ie human readable) form, under a licence that allows it to be modified and redistributed. In addition, many open source licences are "infectious". This means that the requirement to provide source code and permit modification applies to any redistribution of the original software, and even to software that merely contains the original, integrates with the original, or simply uses the original in certain ways.
- 5 There are many different types of open source licence. Each is worded differently and can pose slightly different legal risks. But in general terms the legal risks of these licences can be summarised as follows:

Legal risk	Relevance
Exposure to faults and intellectual property claims.	Relevant to all open source use.
Disclosure of confidential code.	Relevant where software has been infected by an open source licence.
No rights to use.	
Inability to commercialise.	Seldom relevant to government agencies and not covered in this guide.

- 6 It is the infectiousness of open source licences that leads to many of these risks. Unfortunately it is not always clear-cut when any piece of open source software will be infectious. In addition, the practical significance of these risks for any particular piece of software will depend on the intended

use of the software and whether anyone is likely to seek to enforce the terms of the open source licence.

- 7 Managing open source software risks can be complicated. To help simplify matters, SSC makes the following general recommendations to cover most open source legal risks facing government agencies:

7.1 *Using stand-alone, open source applications:*

- (a) Only use open source licences that have been legally reviewed, including the GPL, LGPL, CAL, MBSD, MIT which have been reviewed and are recommended by SSC for use in accordance with this guide.
- (b) Obtain performance and intellectual property warranties from the supplier of the open source software, where appropriate and available.

7.2 *In-house modification or integration of open source software:* In addition to the above recommendations:

- (a) Choose one of the following distribution strategies for the resulting software:
  - (i) Closed distribution, ie only within the agency's legal entity.
  - (ii) Limited distribution, ie to other legal entities on non-open source terms.
  - (iii) Open distribution, ie on open source terms.
- (b) Manage the chosen licence to match the chosen distribution strategy as follows:

<b>Licence</b>	<b>Open distribution</b>	<b>Limited or closed distribution</b>
GPL	May use	Quarantine
LGPL	May use	Quarantine or meet LGPL exception
CAL	May use	Quarantine or meet CAL exception
MBSD	May use	May use
MIT	May use	May use

- (c) Avoid the conflict of licence terms that can occur when more than one open source licence infects the same piece of software.

7.3 *Using third party developers:*

- (a) As the standard contractual position, prohibit use of open source software in all development contracts.
- (b) If the developer wishes to use open source software, consider all of the above recommendations before agreeing to remove the open source prohibition. Include specific contractual provisions in the development contract to ensure the proposed use of open source software is appropriate.

7.4 *Redistributing software:* In addition to the above recommendations, meet the distribution requirements of any relevant open source licences.

8 Agencies should put in place policies and procedures to ensure that these recommendations are met. As the above recommendations won't always fit the situation at hand, agencies should allow for exceptions on a case-by-case basis.

9 SSC's recommended approach can be summarised as follows:

Open source use	Stand-alone applications	Modifying or integrating	Third party developers	Distributing
Only use approved licences	✓	✓	✓	✓
Obtain warranties where relevant	✓	✓	✓	✓
Choose distribution strategy		✓	✓	✓
Manage the licence appropriately		✓	✓	✓
Avoid licence conflicts		✓	✓	✓
Prohibit open source in development contracts			✓	✓
Permit with applicable contract provisions			✓	✓
Comply with distribution provisions				✓

## **PART C – BACKGROUND**

### **KEY FEATURES OF OPEN SOURCE LICENCES**

10 For the purposes of this guide, the term "open source" describes software licensing arrangements that share the following features:

- 10.1 *Modification permitted and source code provided:* Open source licences let licensees use and modify the software's source code.
  - 10.2 *Distribution permitted:* Open source licences allow licensees to distribute the software, whether in its original form, as part of another software program, or as modified by the licensee.
- 11 Open source licences commonly include a number of additional features that are relevant to this guide, including:
- 11.1 *"Infectious" nature:* Many open source licences are "infectious", meaning that the original open source licence may apply to:
    - (a) the original software if re-distributed;
    - (b) any modification of the original software if redistributed;
    - (c) software containing or integrated with the original software, if redistributed; or
    - (d) software used in conjunction with the original software to provide a web based service.

We prefer to use the descriptive term "infectious", in a similar manner to Greg Vetter's widely referenced paper on open source licenses: <http://opensource.mit.edu/papers/vetter2.pdf>.

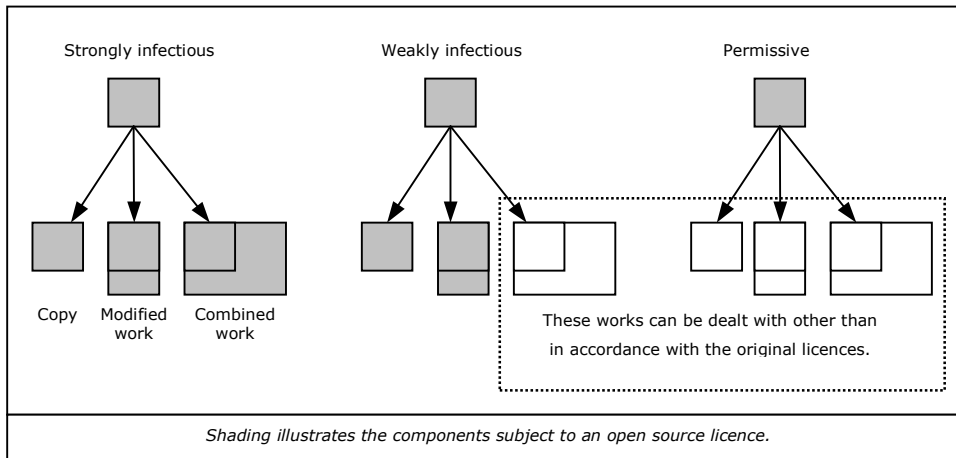
- 11.2 *No warranties:* Open source software is generally provided "as is", without any warranties as to its fitness for a purpose, performance, title or infringement, and without any indemnities against third party claims of intellectual property infringement.
- 12 The Open Source Initiative (OSI) has currently approved over 55 open source licences, including what they describe as the "classic" GPL, LGPL, BSD and MIT licences. As each open source licence is written on different terms, each licence may have different legal effect. As OSI points out, "be sure that you read and understand the license terms completely".

## **UNDERSTANDING THE INFECTIOUS EFFECTS OF OPEN SOURCE LICENCES**

- 13 A key to understanding the legal risks of open source is understanding the extent to which open source licences are infectious.

## ***There are three broad levels of infection***

- 14 Open source licences can be grouped into three broad levels of infectiousness, illustrated below:



### ***Strongly infectious licences***

- 15 A "strongly infectious" open source licence will infect any redistributed piece of software that contains or is derived from software licensed under it. It is generally very difficult to modify or integrate software licensed under a strongly infectious open source licence without the resulting product, when redistributed, becoming "open source" on the same terms as the original. The GPL is an example of a strongly infectious open source licence.
- 16 Software may also be infected by a strongly infectious licence in the following circumstances:
- 16.1 *Infected software output:* It can be argued that any output from a piece of open source software is "derived" from that software, and accordingly is infected. Open source compiler programs are an interesting case in point. A **compiler** translates source code into object code that can be executed on a computer. The object code is the output, and derivative, of the compiler. So any application that has been compiled by a strongly infectious open source compiler may itself be infected. The GPL expressly provides that software compiled with the GNU Compiler Collection (GCC) is not infected by the GPL. Presumably the Free Software Foundation considers other GPL compilers will infect the compiled software.
- 16.2 *Libraries:* A **library** is a collection of subprograms which provide services to independent applications by way of "links". Linking allows library code and data to be shared and changed in a modular fashion between applications. Linking may be static (where the library data is copied into the application when it is compiled) or dynamic (where the library remains distinct from the application). It is not settled whether static or dynamic linking of a strongly infectious open source

library will infect the resulting application. But the GPL assumes they do. Most operating systems include libraries used by the applications that run on them. An exception to the GPL provides that the distributed source code need not include anything normally distributed with the major components of an operating system, ie the libraries that come with the operating system.

16.3 *Programs that communicate with open source applications:* Many programs send instructions to, or receive instructions from, open source applications, including plug-ins, device drivers, clients and GUIs. A **plug-in** is usually a program that interacts with an operating system's user interface, to provide a specific function. Examples are plug-ins to display specific graphic formats (eg, SVG if the browser doesn't support this format natively), to play multimedia files, to encrypt/decrypt email (eg, PGP), or to filter images in graphic programs. A **device driver** is usually a program that enables an operating system to interact with a hardware device by way of an abstraction layer built into the operating system. A **client** is a program that accesses a service on another computer, via a network. A **GUI** or graphical user interface is the means of interacting with a computer by graphic images, such as the windows, icons, menus and pointing devices of most modern operating systems. It has been argued that if these programs are written with specific open source software in mind, they will be infected by the relevant open source licence, but if they were written generally, without knowledge of the internal working of the open source software, then they aren't infected. The legal position is unsettled.

16.4 *ASPs and web services:* An application service provider (**ASP**) is a business that provides computer-based services to customers over a network. A **web service** is a software system designed to support interoperable machine-to-machine interaction over a network. In both cases, the software itself is never distributed. For that reason, open source licences that require distribution in order to become infectious, will have no effect. To counter this, many strongly infectious open source licences include a "network use" clause deeming use over a network to be a distribution. A network use clause may be included in the next version of GPL.

#### *Weakly infectious licences*

17 A "weakly infectious" open source licence will infect only itself and modified or augmented versions of itself; that is, versions which have been added to or changed. Where a developer takes components licensed under a weakly infectious open source licence and integrates them into other software, that other software is not infected, and does not become open source. Note however that this does not necessarily apply in reverse – where the *parent application* is open source under a weakly infectious licence, and a non-

open source component is added to it, more often than not the resulting product will be infected. Again, it all depends on how the licence is worded. The CAL is an example of a weakly infectious open source licence.

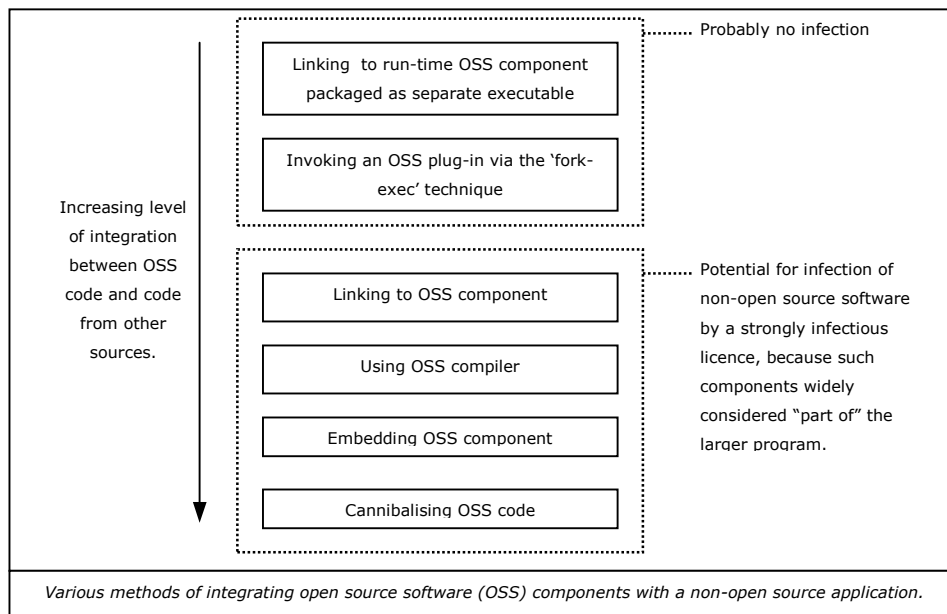
*Permissive licences*

- 18 A "permissive" open source licence is not infectious. A permissive licence does not apply to the original software itself, modifications of the software, or to other software that is integrated with it. The MIT Licence is an example of a permissive open source licence.

***Infectious licences can be quarantined by various integration techniques***

- 19 The infectious effects of open source licences can be "quarantined" by keeping open source components sufficiently separate from the rest of the software. Take for example an application which utilises a component licensed under the GPL (a strongly infectious licence). If this component is integrated with the rest of the application in such a manner that the application as a whole cannot be said to "contain" the component, the package is not infected by the GPL. The developer of the package would have to distribute the component itself under the GPL, but could treat the rest of the package as its own proprietary product.

- 20 The infectious impact of various integration techniques is illustrated below.



- 21 However, the diagram above should be taken as a guide only. There will often be considerable uncertainty over what degree of separation is sufficient to avoid infection by a particular open source licence, because of the following:

- 21.1 Differently worded open source licences require different degrees of separation for successful quarantining. In each case it is a question of interpretation, often requiring legal advice.
  - 21.2 The law that governs an open source licence will influence its interpretation. The governing law may be specified in the open source licence, or it may need to be inferred from the circumstances, including where the transaction took place and the country of residency of the parties. These issues can be complicated, and legal advice from lawyers in the governing jurisdiction may be needed.
  - 21.3 Many open source licences are drafted in language which is ambiguous or even downright vague – phrases such as “derivative” and “separate works” do not provide any clear indications as to which methods of integration (e.g. static linking, dynamic linking) will result in sufficient separation and which will not. “Derivative” is a term originating under US copyright law – US courts have determined that to be a “derivative”, software must be substantially similar to and in some form include a portion of the original work. Derivative may or may not have the same meaning under NZ law.
  - 21.4 To compound the problem, there are virtually no judicial precedents on this point, even in relation to common licences like the GPL. As a result, it is difficult to discern the legal view (as opposed to views of the open source community) on the various methods of integration.
- 22 As a rule of thumb:
- 22.1 There is no infection for “mere use” of open source software. This includes running a standalone open source word processing or spreadsheet package.
  - 22.2 But there is likely to be infection where open source software is modified or integrated into other software. “Integrated” can mean anything from data-sharing between applications to transplanting code from one piece of software to another.
- 23 Because of the uncertainty surrounding this issue, if the consequences of infection are severe for any intended use of a software application, then the approach taken should be conservative and seek to avoid any possibility of infection. There is no substitute here for case-by-case analysis and building up a body of precedents as risk assessments are done.
- ## **LEGAL RISKS IN USING OPEN SOURCE SOFTWARE**
- 24 This section describes the key legal risks of using open source software, and whether these risks are substantial in practice.

## ***There are a number of legal risks***

### *Exposure to faults and intellectual property claims*

- 25 There is a risk that open source software contains functional defects, or breaches a third party's intellectual property rights (e.g. where it contains code misappropriated from proprietary software or functionality in breach of a patent). The absence of warranties and indemnities in most open source licences means the licensee bears this risk. This can be contrasted with the protection usually available under commercial software licences.
- 26 Although patents only protect against unlicensed use in the country in which the patent was granted, patents can create particularly severe intellectual property infringement risks. This is because patents take precedence over an otherwise valid open source licence. In addition, the impact of patents is constantly increasing as the number of software patents increases. Many multinational software corporations maintain large patent portfolios in a range of countries which they can enforce against software developers and users in those countries.
- 27 It is worth noting that although an open source licence may purport to disclaim all warranties, by virtue of the Consumer Guarantees Act 1993 certain warranties, including warranties governing quality and fitness for purpose, may still apply to open source software that was not acquired for the purposes of a business.

### *Disclosure of confidential code*

- 28 Agencies will want to keep some software strictly confidential. Security related software may be an example. There are several circumstances in which confidential software infected by an open source licence might become openly available against the agency's wishes, for example:
- 28.1 If software is infected by an open source licence, an agency may have no right to distribute the infected software unless it does so on open source terms. The agency is in an unenviable position. It must either breach the open source licence or make the confidential source code open.
- 28.2 Because an agency may simply choose not to distribute the confidential, infected software, perhaps a more likely risk is that someone receiving an inadvertently disclosed copy of infected software will have an implied licence to use it on the terms of the infecting open source licence. Because of that implied licence, the agency may be unable to obtain an injunction to stop the recipient using the infected software.
- 28.3 Further, while most open source licences do not *require* redistribution of the source code or any modifications to it, it is conceivable that an

open source licence might require distribution of modifications as a condition of the right to modify and use the original software.

*No rights to distribute infected software*

29 An agency will have no right to distribute:

29.1 infected software on non-open source terms; or

29.2 software that is infected by more than one open source licence, if the open source licences are not compatible (ie each open source licence may require the infected software to be used or distributed under its licence terms and not the other/s).

30 Apart from not meeting its legal obligations, in these situations the agency faces an increased risk that it may need to protect its licensees against third party claims they can't validly use the infected software. This obligation would normally arise under the "IP Warranty" clause in the distribution agreement.

*Inability to commercialise*

31 The above risks can obviously make it difficult to commercialise infected software . As this is not usually an issue for government agencies, commercialisation issues are not dealt with directly in this guide.

*Other risks*

32 Because most open source licences deal only with licensing issues, many important provisions such as taxation, confidentiality and dispute resolution may be missing. There is also the risk, when using open source software under an obscure open source licence, of unusual licence provisions that have not been considered in this guide.

***Risks are affected by the software's intended use***

33 The risks described above do not apply equally to all intended uses of open source software. For example:

33.1 The risks are high if an agency is using open source software in confidential software intended only for internal use;

33.2 The risks are low if the agency is using an open source component in an application that is itself intended for release on open source terms.

***And the licence terms may never be enforced***

34 Apart from the risks of exposure to faults and third party intellectual property claims, the other risks described above will only eventuate if a

third party enforces the open source licence against the licensee. So who would do that?

*Legal proceedings can only be taken by certain people.*

- 35 The only parties entitled to take *legal proceedings* in relation to open source software are the immediate licensor (for breach of the open source licence) and the owner of the code (for breach of copyright if the open source licence is not followed). Legal proceedings are expensive, and there is likely to be little incentive on the direct licensor or the owner of the code to enforce their rights legally – after all they must have been happy for their code to be freely used and modified or they would not have released it on open source terms.
- 36 Enforcement is complicated further by the fact that open source software can have multiple owners, as the author of each part of the software may own that part. To address this issue, some open source organisations encourage authors to assign ownership of any new open source code to the open source organisation. This concentrates ownership in one organisation and simplifies any enforcement of the open source licence by that organisation.

*Who might enforce a claim?*

- 37 So there are probably few situations in which anyone would enforce an open source licence. But there are some situations where they may, including the following:
- 37.1 *Defence from unlicensed user:* If software is infected by an open source licence (because it contains open source code), the terms of the open source licence could legitimise use that would otherwise be unauthorised. In other words, an unauthorised user might claim as a defence that the owner should have licensed the software on open source terms.
- 37.2 *Claim by interest group:* In some circumstances, an open source interest group, such as the Free Software Foundation, may fund a claim on behalf of the code owners, or take a claim itself where it is the owner or direct licensor. It has been reported that, in 2003, the Free Software Foundation took out-of-court action against some 50 GPL infringers.
- 37.3 *Public pressure:* Perhaps a more likely approach, although not a legal one, is that the open source community may put political pressure on the infringing organisation to comply with the open source licence.

## **PART D – RECOMMENDATIONS**

- 38 Dealing with the risks of open source software can be complicated, given the multitude of open source licences, the ways in which they can infect other software, and the numerous ways in which open source software may be used. To help simplify matters, this guide contains recommendations that SSC hopes will address the risks in most open source use by government agencies.
- 39 The recommendations cover the main situations in which government agencies might acquire and use open source software, namely:
- 39.1 Licensing stand-alone open source applications for internal use.
  - 39.2 Internal development using open source software.
  - 39.3 Using contractors to develop software.
  - 39.4 Distributing software to third parties.
- 40 Government agencies should put in place policies and procedures to ensure that the recommendations are met. As the recommendations are necessarily general, and err on the side of caution, agencies should implement a process for granting appropriate exemptions to the recommendations on a case-by-case basis.

## **USING STAND ALONE OPEN SOURCE APPLICATIONS**

- 41 If an agency is merely using open source software, there is usually no risk of infection and the agency should follow the recommendations in this section.
- 42 However, as illustrated in the section headed "Understanding the Infectious Effects of Open Source Licences", it can be unclear what constitutes "mere use" without infection. In inherently risky situations (eg when using highly confidential software together with the open source application to provide web services) the agency may want legal advice on the likelihood of any open source infections.

### ***Only use specific open source licences***

- 43 Because of the complexities of the many different open source licences, SSC only recommends use of open source software under the following licences:
- 43.1 the GNU General Public Licence (GPL), a strongly infectious licence;

- 43.2 the GNU Lesser General Public Licence (*LGPL*), a strongly infectious licence containing exceptions allowing it to act like a weakly infectious licence if certain criteria are met;
  - 43.3 the Clarified Artistic Licence (*CAL*), a weakly infectious licence;
  - 43.4 the Modified BSD Licence (*MBSD*), a permissive licence; and
  - 43.5 the MIT Licence (*MIT*), a permissive licence.
- 44 If an agency proposes to use open source software under any other open source licence, it should seek approval under Part E – Exceptions.

***Obtain warranties/indemnities if appropriate and available***

- 45 The warranties and indemnities that an agency should seek from external suppliers, where appropriate and available, include the following:
- 45.1 A warranty that the software conforms to the supplier’s specification and the agency’s requirements. Suppliers will generally be unwilling to provide a more open-ended “fitness for purpose” warranty.
  - 45.2 A warranty that the agency’s use of the software in accordance with the agreement will not breach the intellectual property rights of any third party.
  - 45.3 An indemnity from any third party’s claim that its intellectual property rights have been infringed by the agency using the software in accordance with the agreement.
- 46 These warranties and indemnity should protect against patent claims in all jurisdictions in which the code may be used. Their effectiveness will depend on the precise terms of the warranties and the supplier’s financial means. It is good practice to have the warranties legally reviewed.
- 47 It will be appropriate to obtain these warranties and indemnities if:
- 47.1 the agency is paying a significant amount of money for the software or a wider package which relies on it;
  - 47.2 the software is to be used in a “mission critical” application, i.e. where a failure would cause serious loss or disruption; or
  - 47.3 the supplier of the software recommended the use of the open source software as an alternative to a proprietary product.

- 48 Whether the agency will be able to negotiate warranties and indemnities in respect of open source software will depend on its negotiating leverage with the supplier.
- 49 Where the agency is *not* able to obtain suitable warranties and indemnities, it should consider alternatives to the open source software in question. Proceeding without relevant warranties and indemnities does not require sign-off under Part E – Exemptions, but before proceeding the agency should weigh:
- 49.1 the benefits of using the particular open source software – for example, cost savings or the ability to customise; against
- 49.2 the likelihood and amount of loss which might be sustained by the agency, for example:
- (a) lost productivity, data loss or repair costs, in the event of a technical failure; and
- (b) negative publicity, legal expenses or damages for infringement, if the agency is accused by a third party of infringing its intellectual property rights.

## **INTERNAL MODIFICATION AND INTEGRATION**

- 50 If an agency is modifying/integrating open source software, it should treat the open source licence as infectious and follow the recommendations in this section, in addition to the recommendations set out in the section above. If the open source licence is not infectious, only the recommendations in the section above need be followed.

### ***Choose a distribution strategy***

- 51 The government agency should first choose an appropriate distribution strategy for the finished product. There are three possible strategies:
- 51.1 *Closed distribution:* Because of the nature of the software, eg its strict confidentiality, it must never be distributed outside the legal entity in which it originates.
- 51.2 *Limited distribution:* The software may be distributed outside the legal entity in which it originates, now or in the future, but the agency will want to restrict distribution to a limited group of recipients and/or licence the software on non-open source terms, eg without providing the source code.
- 51.3 *Open distribution:* The agency is comfortable that:

- (a) if it wished to distribute the software, it would do so on open source terms; and
- (b) inadvertent release of the software would not unduly prejudice the agency.

52 It is important to note that government departments are part of the single legal entity, the Crown. Accordingly, a closed distribution strategy for one department would permit distribution amongst all departments. However Crown Entities (EQC or NZQA, for example) and SOEs are not part of the Crown and are distinct legal entities from each other. So distribution between these entities would require a limited distribution strategy rather than closed.

53 It should also be noted that an open distribution strategy does *not* mean that the agency must distribute the software. An agency must consider its obligations under the Official Information Act 1982 if it receives a request to disclose software it owns or has licensed.

***Manage the chosen licence to match the chosen strategy***

54 Once the government agency has chosen a distribution strategy and the open source licence it will use, it should manage its use of the open source code in accordance with the following table.

<b>Licence</b>	<b>Open distribution</b>	<b>Limited or closed distribution</b>	
GPL	May use	Quarantine	
LGPL	May use	Quarantine or meet LGPL exception	
CAL	May use	Quarantine or meet CAL exception	
MBSD	May use	May use	
MIT	May use	May use	

55 The requirements set out in the above table are explained in more detail in the following paragraphs.

*Quarantining GPL code*

56 Quarantining is necessary where the agency wants to develop or commission software utilising GPL code, but the software is intended for **limited** or **closed** distribution. In other words, quarantining is a method of *avoiding* the infectious terms of the GPL in order to guarantee limited or closed distribution.

57 It is a somewhat risky approach. The GPL is not entirely clear on the degree or methods of separation necessary to prevent the GPL from

infecting software which utilises code or components licensed under it. However, we believe it is safe for an agency to quarantine GPL code by confining the code to a separate executable or a plug-in invoked by a technique such as "fork-exec" (in Unix). Any other form of integration requires a risk assessment and legal sign-off on an exception basis as set out in Part E - Exceptions.

- 58 Where GPL code is successfully quarantined, the rest of the software can be distributed under proprietary terms, and without providing the source code. Note however that the quarantined code (for example, the plug-in) must still be distributed under the terms of the GPL.

*Quarantining or meeting LGPL exception*

- 59 Where the agency chooses a **closed** or **limited** distribution strategy, code licensed under the LGPL will be suitable for modification or incorporation into other software **only if** one of the following apply:

59.1 First, the LGPL code may be quarantined in the same manner as for GPL software, described above.

59.2 Secondly, under clause 6 of the LGPL, software that uses a LGPL library may be distributed without source code if the LGPL library is linked by a mechanism which:

- (a) uses at run time a copy of the library already present on the user's computer system (ie a dynamically linked library); and
- (b) will operate properly with a modified version of the library if the user installs one, as long as the modified version is interface-compatible with the version that was originally linked to by the application.

Before utilising this exception to the LGPL, those responsible for the software should consult the text of the LGPL, particularly as clause 6 includes other distribution requirements such as permitting modification for the customer's own use (even though source code is not provided).

*Quarantining or meeting the CAL exception*

- 60 Where the agency chooses a **closed** or **limited** distribution strategy, code licensed under the Clarified Artistic Licence will be suitable for modification or incorporation into other software **only if** one of the following apply:

60.1 First, the CAL code may be quarantined. The CAL is weakly infectious as it only applies to the original files and derivatives of the original files "that are created through textual modification". Accordingly, it is safe to link to CAL software without risk of infection.

60.2 Secondly, clause 4.3 of the CAL provides a way to avoid such derivatives being infected. The clause requires the distributor to:

- (a) renames any non-standard executables so the names do not conflict with names of the original executable;
- (b) document in a manual how each executable differs from its original version; and
- (c) provide instructions on where to get the original version to anyone to whom it distributes the software.

61 Before utilising this exception, those responsible for the adaptation of the open source software should consult the Clarified Artistic Licence, particularly clauses 3 and 4 and its additional requirements for notice of modifications.

### ***Avoid conflicts of open source licences***

62 Care should be taken to avoid licence conflicts which may occur when a piece of software is infected by two or more different licences with incompatible requirements; that is, where complying with one licence would result in a breach of the other. The licences listed above are widely accepted as compatible with the GPL and each other.

## **THIRD PARTY SOFTWARE DEVELOPMENTS**

63 When using third party developers, agencies have less control over whether open source software is included in the developed software, and hence there is often a higher risk of infection. To address these risks, the agency should follow the recommendations in this section.

### ***Prohibit use of open source software without consent***

64 As its standard position, all Development Agreements should prohibit the use of any open source code in the supplied software,

***If open source used and approved, include specific contractual provisions***

- 65 If the developer wishes to use open source software, the agency may remove the prohibition only after considering all of the recommendations above.
- 66 If the agency gives its consent for the third party to include open source code in the software, the following provisions should be included in the development contract, depending on the desired distribution strategy:

Open distribution	Limited distribution	Closed distribution
<ul style="list-style-type: none"> <li>• Specify precisely what open source software is being used, and which open source licence(s) applies.</li> <li>• Specify whether the original source code is licensed to the agency by the developer, or whether the agency must obtain its own licence.</li> </ul>		
<ul style="list-style-type: none"> <li>• Vest, in the developer or the agency, all intellectual property rights in any new code created by the developer.</li> <li>• If ownership rests in the developer, require the developer to license all the new code to the agency on the terms of the applicable open source licence.</li> </ul>	<ul style="list-style-type: none"> <li>• Vest in the agency all intellectual property rights in any new code created by the developer.</li> <li>• Provide that the developer will only access the new code as the agent of the government agency.</li> <li>• Require the developer to keep the new code confidential.</li> <li>• Where appropriate, specify how open source components will be quarantined from the rest of the software.</li> </ul>	

- 67 If the new code is owned by an agency that is part of the Crown, it will be subject to crown copyright. Otherwise it will be subject to ordinary copyright. There is little difference between the two types of copyright for the purposes of this guide.

**SOFTWARE DISTRIBUTION**

- 68 If an agency wishes to distribute any software, it should follow the recommendations in this section. The aim is to ensure that if the software has been infected, that the underlying open source licences are complied with.

***Confirm whether any open source licence applies***

- 69 The first step is to confirm what open source licences apply. It is not always an easy task. One approach is to identify who developed each module and determine on what basis the agency is using it (eg as owner or licensee and, if as licensee, then on what terms). Another approach is to read the End User License Agreement and/or search the source code for copyright notices, to identify if any open source code can be found.

70 The next step is to identify whether the open source code has infected any other software being distributed.

***Meet the distribution requirements for all infected software***

71 For any infected software, the distribution requirements of the relevant open source licence should be complied with. The agency should consult the relevant open source licences to confirm the distribution requirements. They are likely to include:

71.1 Providing copies of source code

71.2 Including copyright notices. The licences approved under this guide require the following copyright notices and disclaimers with any infected software:

Licence	Distribution requirements
GPL	(a) Insert prominent notices into any modified files stating that the agency changed the files, and specifying the date of the changes.  (b) Carry over any copyright notices and disclaimers which were displayed on running the original version, (including instructions on how to obtain to a copy of the GPL) and ensure that these display on running the modified version.
LGPL	If the parent application displays any copyright notices, include a copyright notice for the library, including instructions on how to obtain a copy of the LGPL.
MBSD	Include the copyright notice and associated conditions and disclaimers contained in the original licence.
MIT	Include the copyright and permission notices contained in the original licence.

**PART E - EXCEPTIONS**

72 An agency may wish to depart from the recommendations in this guide in a number of instances, including:

72.1 to use an open source licence that isn't approved for use under this guide; or

72.2 to use a specific form of integrating open source software, in order to avoid infection by that software.

73 Where an agency seeks to use an exception, it should clearly understand the risks and weigh them against the benefits of the proposed approach. The staff who are seeking the exception should complete the questionnaire below. With this information, the agency's legal advisors and other relevant experts can assess and approve the except as appropriate.

Issue	Question
Infection	What is the name of the open source licence which applies to the software?
	Will you modify the open source software in any way?
	Will the open source software be used in conjunction with any other software? If so: <ul style="list-style-type: none"> <li>How will the two be integrated?</li> <li>Will the open source components be easily removable?</li> </ul>
Distribution	What is the proposed distribution strategy?
	Will you be distributing the open source software outside of the agency? If so, to whom?
Risks	Will the open source software help provide a critical business function, or will it only provide minor functionality?
	What is the potential impact of any unauthorised use?
Alternatives	Are there commercial software products which offer the same or similar functionality at an acceptable price?
	Would it be difficult or costly for the agency to write its own software to provide the same or similar functionality?
Warranties	If a vendor is supplying the open source software for a fee, what is the agency's negotiating position? Is the vendor likely to offer: <ul style="list-style-type: none"> <li>a warranty as to performance; or</li> <li>an indemnity for any breach of third party intellectual property caused by our use of the software?</li> </ul>

## PART F – EXAMPLES

- 74 To illustrate the application of this guide, this section contains a number of examples.

### EXAMPLE ONE - STAND ALONE APPLICATION

- 75 In this example, an agency wishes to license a stand-alone open source application under the GPL, without any warranties. The application is to be obtained free of charge via the web.

Recommendation	Application	Exception
Only use approved licences	GPL used.	NA
Obtain warranties where relevant	Warranties not expected for free-of-charge application.	NA

### EXAMPLE TWO - INTERNAL DEVELOPMENT

- 76 In this example, staff at a government department wish to use GPL components in a strictly confidential application, that will be distributed to other government departments and agencies. The components are available free of charge via the web, and will be dynamically linked to the application.

Recommendation	Application	Exception
Only use approved licences	GPL used.	NA
Obtain warranties where relevant	Warranties not expected for free-of-charge application.	NA
Choose distribution strategy	Limited distribution.	NA
Manage the licence appropriately	NA	Dynamic linking requires specific risk assessment and legal signoff.
Avoid licence conflicts	No conflicts, as only one open source licence.	NA

### EXAMPLE THREE - EXTERNAL DEVELOPMENT

77 In this example, an external developer wants to use open source software under the GPL and a non-approved licence. The resulting application will be released by the agency on open source terms. The agency is paying market rates for the application. The developer wishes to use the new code for other customers.

Recommendation	Application	Exception
Only use approved licences	NA	Legal review shows alternative licence is weakly infectious and GPL compatible.
Obtain warranties where relevant	Warranties expected, as market rates paid.	NA
Choose distribution strategy	Open distribution.	NA
Manage the licence appropriately	Free to use.	NA
Avoid licence conflicts	No conflict, as licences compatible.	NA
Prohibit open source in development contracts	Consent given.	NA
Permit with applicable contract provisions	Permit GPL and other licence only. Developer owns new software, and licenses new and existing software to agency under GPL. Warranties apply equally to all licensed software.	NA
Comply with distribution provisions	Some redistribution requirements.	NA

**END.**